Submitted to PAKDD 2005:
Paper ID: #223

Considering Re-occurring Features in Associative Classifiers

Rafal Rak, Wojciech Stach,
Dept. Electrical and Computer Engineering
University of Alberta, Canada
{rrak, wstach}@ece.ualberta.ca

Osmar R. Zaïane, Maria-Luiza Antonie
Department of Computing Science
University of Alberta, Canada
{zaiane, luiza}@cs.ualberta.ca

Abstract

The classification problem is one of the most common tasks in Data Mining and Machine Learning. Given its vast applicability in many real domains, supervised classification has been addressed and extensively studied. There are numerous different classification methods; among the many we can cite associative classifiers. This newly suggested model uses association rule mining to generate classification rules associating observed features with class labels. Given the binary nature of association rules, these classification models do not take into account repetition of features when categorizing. Repetitions of features are often good indicators and discriminators of classes, in particular for text or other multimedia. In this paper, we enhance the idea of associative classifiers with associations with re-occurring items and show that this mixture produces a good model for classification when repetition of observed features is relevant in the data mining application at hand.

1 Introduction

Classification is one of the most common tasks in data mining and machine learning. By and large, it consists of extracting relevant features from labelled training data to build a model that discriminates between classes for unlabelled observed objects. Myriad techniques have been proposed and while there are, in general, better approaches than others, there is no clear winner in terms of correctness and usability given a particular problem application. Among the numerous different classification methods [4] we can distinguish those providing a model in the form of rule sets, e.g. decision trees, rule learning, or naïve-Bayes. These approaches have several benefits that come from having rules that describe the classification model. One of the most important advantages is that such a model is *transparent*, i.e. experts from the domain of the application are able to understand it and ultimately edit it. This feature allows them to manipulate and add rules in order to increase the confidence and accuracy of the classifier. Amid these rule-based classification models is the associative classification model.

Associative classification is a relatively new method. The main objective is to discover strong patterns that are associated with the class labels in the training set. The training set is modeled into transactions with items being the observed features. As a final classification model, one obtains a set of association rules associating features with class labels. In the literature, there are few known classifiers based on the above-mentioned idea, i.e. CBA [7], CMAR [6], and ARC-AC/ARC-BC [12].

One considerable limitation of all these algorithms is that they do not handle the observations with repeated features. In other words, if a data object is described with repeated features, only the presence of the feature is considered, but not its repetition. However, in many applications such as medical image categorization or other multimedia classification problems, the repetition of the feature may carry more information than the existence of the feature itself [13]. For example, the appearance of two particular lesions of given type in a brain scan is more indicative than the mere presence of the lesion type [13]. Also in text mining and information retrieval, it is widely recognized that the repetition of words is significant and symptomatic, hence the common use of TF/IDF (i.e. the frequency of a term in a document relative to the frequency of the term in a collection).

Associative classifiers use association rule mining to build a classification model. However, association rule mining typically considers binary transactions; transactions that indicate presence or absence of items. No matter how many loaves of bread were bought, a transaction indicates only the presence of bread in the cart and thus the discovered association would be between the presence of bread and the presence of other items regardless of the number of times bread is repeated in the same transaction. Binary transactions simply do not model repetitions. There are numerous applications for which the consideration of the number of the occurrences of items (e.g. similar objects in the same medical image) might be more beneficial than presence or absence of items. A few approaches to mining association rules with re-occurring items have been proposed, such as MaxOccur [13], FP'-tree [9] and WAR [10].

The main goal of our research is to devise a classifier that combines the idea of associative classification and association rules with reoccurring items. Our contributions presented in this paper exploit, combine, and extend the ideas mentioned above, especially ARC-BC and MaxOccur algorithms. We also suggest new strategies to select rules for classification from the set of discovered association rules.

A delicate issue with associative classifiers is the use of a subtle parameter: *support*. Support is a difficult threshold to set, inherited from association rule mining. It indicates the proportion of the database transactions that support the presence of an item (or object). It is known in the association rule mining field that the support threshold is not obvious to tune in practice. In the associative classification literature it has been commonly and arbitrarily set to 0.1%. However, the accuracy of the classifier can be very sensitive to this parameter. In the case of re-occurring items, there are two ways of calculating support: transaction-based support and object-based support [13] (i.e. either the proportion of transactions or the proportion of objects that support the existence of an object in the database). Our experiments show that an associative classifier that considers re-occurrence of features is considerably less sensitive to the variation of support. This leads to more practical applications and eventually the possibility to automatically determine and tune this parameter.

The remainder of the paper is organized as follows: Section 2 presents the problem statement: the model of an associative classifier and the consideration in the model of recurrent items. Related work on associative classification and mining association rules with repetitions is presented in Section 3. We present our new approach ACRI in Section 4. The

experiments showing the performance of our approach are presented in Section 5. Section 6 concludes and highlights some future work.

2 PROBLEM STATEMENT

The original approach of classification using association rules, named *class association rules* (CAR), was introduced in [7]. The main idea was to modify the form of transactions known from the traditional approach to the form of $< condset, c >$, where *condset* is a set of items and *c* is a class label. In other words, objects in a training set are represented by sets of features appended with the observed class label. This forms the transactions to mine. All the rules generated from frequent itemsets are of the form $condset \rightarrow c$. This means that the rules are restricted to those with a class label as a consequent. Once the classifier (in this case: set of rules) is found, it can be used to predict the class of new objects. However, two main problems might occur. One of them is that two or more contradictory rules might exist, i.e. rules that have the same *condset*, yet different class labels. This is not acceptable in the case of single-class classification applications, and these contradictory rules are simply eliminated or only the rule with highest confidence is preserved. In the case of multiple-class classification applications, these rules are not considered contradictory and are preserved for their obvious benefit. The other problem concerns situations, in which there is no exact rule, i.e. rule having the same *condset*, for the object being classified. Different strategies can be applied to handle these cases. We point to some strategies in Section 4.2.

Our task is to combine the associative classification with the problem of recurrent items. More formally, it can be stated that our goal is to modify the original approach using transactions from the form of $< \{i_1, i_2, \ldots i_n\}, c >$, where i_i is an item in a transaction (e.g. a word in a document text) and *c* is a class label, to the form of $< \{o_1 i_1, o_2 i_2, \ldots o_n i_n\}, c >$, where o_i is the number of the occurrences of the item i_i in the transaction. In other words, each item is represented by a *(value, attribute)* pair. Hence, we can treat a transaction as a set of *(value, attribute)* pairs and a class label e.g. $< \{(3, A_1), (2, A_2)\}, c >$, where A_1 and A_2 are attributes values. Different notations of this type of transaction can be used. For simplicity, in this paper we use the following one: $< \{3A_1, 2A_2\}, c >$. The rules generated from this set of transaction have the form $< condset, c >$ and they are used for classification of new objects. Our hypothesis is that associative classifiers with recurrent items have more discriminatory power since they maintain and exploit more information about both objects and rules. Moreover, transactions containing repeated items can support the presence of an item more than just once (i.e. given the re-occurrence). This leads to a different notion of support that is not relative to the size of the training set but the repetitions of the observed features of the different objects in the training set, yielding a more stable classification model.

3 RELATED WORK

Association rules have been recognized as a useful tool for finding interesting hidden patterns in transactional databases. Several different techniques have been introduced to tackle this problem effectively. The most important methods are those based on either Apriori [13] or FP-growth [13] approaches. However less research has been done considering transactions with reoccurrence of items. In [10], the authors assign weights to items in transactions and introduce the *WAR* algorithm to mine the rules. This method is two fold: in

the first step frequent itemsets are generated without considering weights and then weighted association rules (WARs) are derived from each of these itemsets. *MaxOccur* algorithm [13] is an efficient Apriori-based method for discovering association rules with recurrent items. It reduces the search space by effective usage of joining and pruning techniques. The *FP'-tree* approach presented in [9] extends the FP-tree design [13] with a combination from the MaxOccur idea. For every distinct number of occurrences of given item, the separated node is created. In case when a new transaction is inserted into the tree, it might increase support count for the different path(s) of the tree as well. This is based on the intersection between these two itemsets. Given the complete tree, the enumeration process to find frequent patterns is similar to that from the FP-tree approach [13].

One of the very interesting and promising applications of association rules is a classification task. Several classifiers have been introduced so far, i.e. CBA, CMAR, ARC-AC, and ARC-BC. However, they use rules without reoccurrence of items in a single transaction. *CBA* [8], an Apriori-based algorithm, labels new objects based on the confidence of matched rules. *CMAR* [6] using the FP-growth algorithm produces strong rules by pruning more specific and less confident rules by more confident and general ones. If more than one rule matches a given object, advanced tests are performed to select the strongest group of rules with the same label. *ARC-AC* and *ARC-BC* [12] are based on the Apriori algorithm. The ARC-BC approach treats each class in the training set separately. That is, it considers each group of transactions labelled by the same category separately while the ARC-AC considers all categories combined. In order to manage the selection of rules, the *dominance factor* is introduced, which is defined as a proportion of rules of the most dominant category in the applicable rules for an object to classify. ARC-AC and ARC-BC were originally developed for classifying text [12] and later on used to classify objects in image collections [2] [3].

4 PROPOSED APPROACH

Based on the amalgamation of our work in associative classifiers and associations rules with reoccurrence, we introduce the new classification method. Our approach, *ACRI (Associative Classifier with Reoccurring Items)*, consists of two modules: Rule generator and classifier. We decided to base our algorithm for mining associations with reoccurring items on Apriori-based MaxOccur. The building of the classification model follows our previous ARC-BC approach. The rational is based on the efficiency of this method in the case of non-evenly distributed class labels. Indeed other associative classification methods are biased towards dominant classes in the case when rare classes exist. Rare classes are classes with very few representatives in the training set. MaxOccur run on transactions from each known class separately makes the core of our *rule generator* module. It mines the set of rules with reoccurring items from the training set. These rules associate a condition set with a class label such that the condition set may contain items preceded by a repetition counter. The discovered rules form the classification model which is used by the *classifier* module. The classification process might be considered as plain matching of the rules in the model to the features of an object to classify. Different classification rules may match, thus the *classifier* module applies diverse strategies to select the appropriate rules to use. In addition, simple matching is sometimes not possible because there is no rule that has the antecedent contained in the feature set extracted from the object to classify. With other associative classifiers, a default rule is applied, either the rule with the highest confidence in the model or simply assigning the

label of the dominant class. Our ACRI approach has a different strategy allowing partial matching or closest matching by modeling antecedents of rules and new objects in a vector space.

4.1 RULE GENERATOR

This module is designed for finding all frequent rules in the form of $<\{o_1 i_1, o_2 i_2, \ldots, o_n i_n\}, c>$ from a given set of transactions, i.e. rules that have support equal or greater than the user-defined *min_support*, the conventional parameter in association rule mining.

The general framework of this part is based on ARC-BC approach [12]. This means that the initial set of transactions representing the training set is divided by categories and rules are generated for each of them independently, see Figure 1.

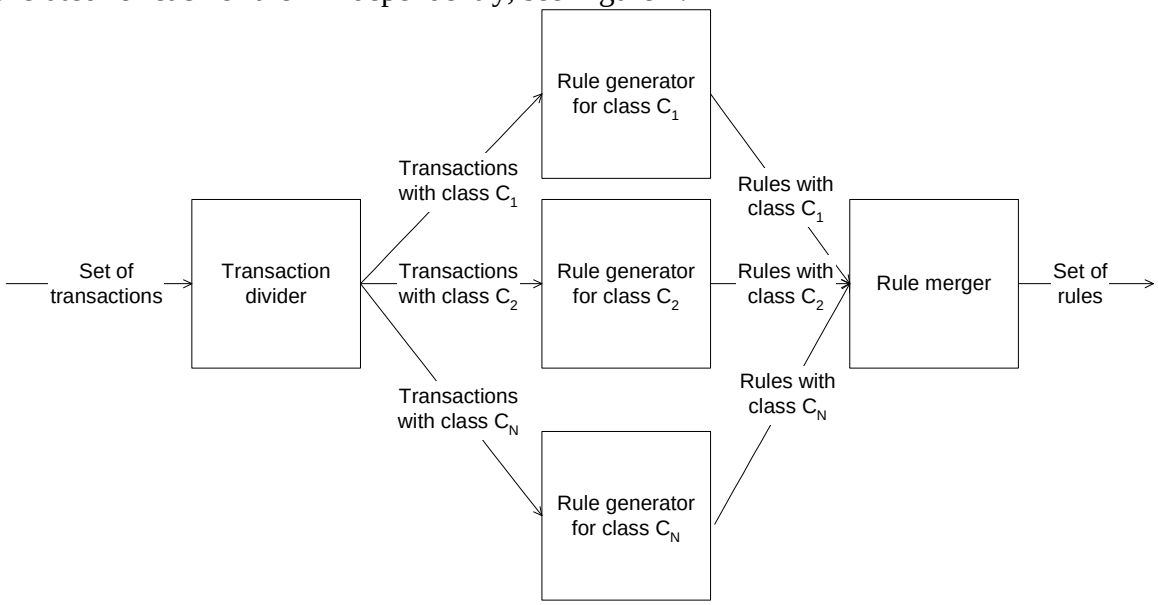

Figure 1 High-level diagram of ACRI rule generator module

The main components of this module are the following:

- *Transaction divider* – this block scans the set of transactions once and creates *N* subsets of this set – each for one category (*N* is equal to the number of classes);
- *Rule generator for class C_x* – this is an Apriori-based algorithm for mining frequent itemsets that extends the original method by taking into account reoccurrences of items in a single transaction à la MaxOccur. In order to deal with this problem, the support count was redefined. Typically, a support count is the number of transactions that contain an item. In our approach, the main difference is that single transactions may increase the support of a given itemset by more than one. We say that transaction *supports* itemset by a given number. The formal definition of this approach, which was proposed in [13] as a MaxOccur algorithm, is given below.

Transaction $T=<\{o_1 i_1, o_2 i_2, \ldots, o_n i_n\}, c>$ supports itemset $I=\{l_1 i_1, l_2 i_2, \ldots l_n i_n\}$ if and only if $\forall i=1...n \;\; l_1 \le o_1 \wedge l_2 \le o_2 \wedge ... \wedge l_n \le o_n$. The number t by which T supports I is calculated according to the formula: $t = \min\left[\frac{o_i}{l_i}\right] \;\; \forall i = 1...n : l_i \neq 0 \wedge o_i \neq 0$

Example: Let us take into account the following transaction $T =<\{2i_1, 4i_2, 6i_7\}, 1>$. Support for several itemsets given by this transaction is shown in Table 1.

Table 1: Example of support counting with ACRI

Itemset *I*	**Support *t***
$\{2i_1, 3i_2, 5i_7\}$	1
$\{i_1, 2i_2, 3i_7\}$	2
$\{i_2, 2i_7\}$	3
$\{i_1, i_2, i_3\}$	0

Thus, the rule generator module finds all the itemsets that are frequent according to the above definition of support.

- *Rule merger* – this block collects the rule sets for different classes and merges them in order to generate a complete set of mined rules. In this part we do not perform any pruning even if there are contradicting rules, thus the merging is nothing but collecting the rules together.

4.2 CLASSIFIER

This module labels new objects based on the set of mined rules obtained from the *rule generator*. An associative classifier is a rule-based classification system, which means that an object is labelled on the basis of a matched rule (or set of rules in case of multi-class classification). This task is simple if there is an exact match between a rule and an object, i.e. *antecedent* of the rule and the object features are identical. The model, however, often does not include any rule that matches a given object exactly. In such a case, in order to make the classification, all rules are *ranked* according to a given scenario and the best one (or several) is matched to a given object. Rule ranking might be performed following different *strategies*, which associate each rule to a number that reflects its *similarity* to a given object. These strategies may be used either separately or in different combinations. We have tested the following ones: cosine measure, distance measure, coverage, confidence, support, and dominant matching class, which are characterized below.

Let us consider the rule $<\{o_1 i_1, o_2 i_2, \ldots, o_n i_n\}, c>$ and the object to be classified $<l_1 i_1, l_2 i_2, \ldots, l_n i_n>$. The corresponding n-dimensional vectors can be denoted as $\vec{o} = [o_1, o_2, ..., o_n]$ and $\vec{l} = [l_1, l_2, ..., l_n]$. The three following measures are based on this representation.

- *Cosine measure (CM)* – assigns a value that is equal to the angle between these two vectors, i.e. $CM = \arccos \angle(\vec{o}, \vec{l})$. The smaller the CM value is, the smaller the angle, and the closer these vectors are in the n-dimensional space. It is equal to zero if the vectors have the same direction, which, roughly speaking means that they have the same "proportions" of items.

- *Distance measure (DM)* – assigns a value that is equal to the distance between the ending points of these two vectors according to a given distance norm, i.e. $DM = \text{distance}(\vec{o}, \vec{l})$. We have tested norm L1 and norm L2 as distance functions. In general, this measure "standalone" seems to be useless, since the vectors might have various lengths. The only rational usage might be as a "fine tuning", i.e. when the set of rule has been already pruned. The smaller the distance is, the closer the two ending points of these vectors are.
- *Coverage (CV)* – assigns a value that is equal to the ratio of the number of common items in the object and rule to the number of items in the rule (ignoring reoccurrences).
  In this case, the larger is the *CV* ratio, the more items are common for the rule and the object. *CV*=1 means that the rule is entirely contained in the object.

The two following ranking methods refer to the rule property only and do not depend on the classified object. Thus, they have to be used with other measures that prune the rule set.

- *Confidence*: From the matching rules, select the rule with best conditional probability of the antecedent knowing the class (i.e. best confidence).
- *Support*: From the matching rules, select the rule with best probability of the antecedent in the class (i.e. best support).

- In the last examined classification scenario, called *Dominant matching class*, the class label is assigned to the object by choosing the one being the most frequent from the set of rules matching the new object. Notice that dominance can be counted by simply enumerating the matching rules per class or a weighted count using the respective confidences of the matching rules.

5 EXPERIMENTS

We tested ACRI on different datasets to evaluate the best rule selection strategy as well as compare ACRI with an associative classifier like ARC-BC. As an example, we report here an experiment with the *mushroom* dataset from the UCI repository [15]. We compare ACRI with ARC-BC later on using the Reuters dataset as used in the paper presenting ARC-BC for text categorization [15].

It appears that the rule selection strategies have roughly similar performance in terms of accuracy. However, this accuracy varies with the support threshold. The lower the support, the more rules are discovered allowing a better result using selection based on cosine measure for example. Using the *dominant matching class* was also doing well, confirming the benefit of the *dominance factor* introduced in [12]. The distance measures (L1 and L2) were not satisfactory in general and are not reported here. The same is true for selection based on best rule support. Results were unacceptable. We also observed that *coverage* (*CV*) gave better results when set to 1. Thus all results reported herein have *CV* set to 1. The other measures are comparable in performance and trend, except for *best confidence*. When the support threshold is high, fewer rules are discovered and confidence tends to provide better results while the cosine measure returns matches that have big angles separating them from the object to classify, hence the lower accuracy. Figure 5 shows the superiority of the rule selection strategy *dominant matching class* up to a support threshold of 25%, beyond which *best confidence* becomes a winning strategy. Figure 6 shows how the more rules are discovered the more effective in terms of accuracy the strategies *dominant matching class* and *cosine*

measure becomes in comparison to *best confidence* approach. The number of rules is correlated with support.

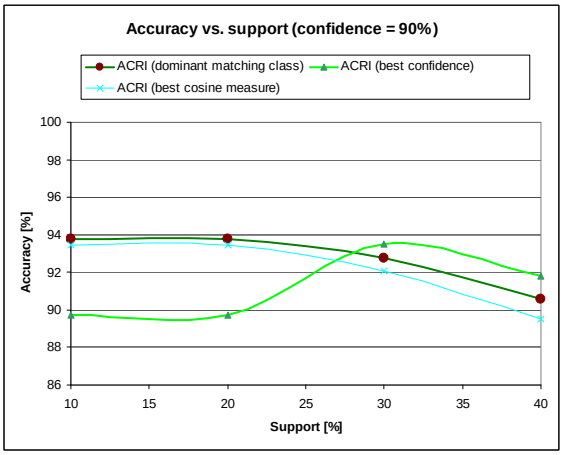

Figure 1: Accuracy of different rule selection strategies

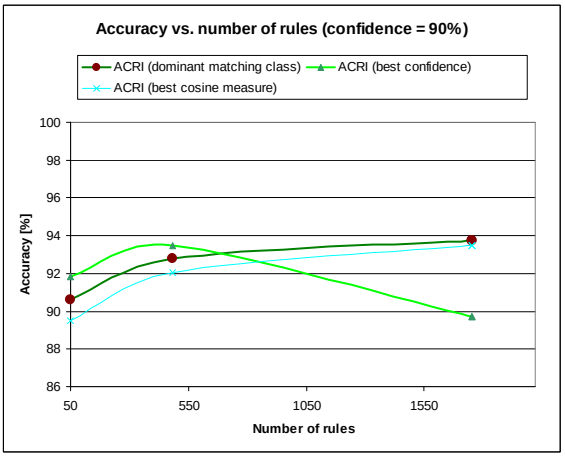

Figure 2 Accuracy vis-à-vis number of rules

5.1 EXPERIMENTAL SETUP FOR REUTERS DATASET

We used the Reuters-21578 text collection to perform comparative experiments. We chose the "ready-to-use" top 10 topics [15] from this dataset. The total of 9980 documents is split into two sets: 7193 and 2787 for a training and test set respectively. First, we pre-processed data extracting text from XML documents. For normalization of words we used Porter's algorithm [14] to stem the words. We also pruned stop words, i.e., words that appear too frequently and do not contribute to the results. The list of stop words was a combination of the list used in [12] and words from our observations while performing tests (e.g., an obvious stop word is the word *Reuters* as it appears in every document and should be treated as noise rather than as useful information).

5.2 COMPARISON TESTS

In order to compare the results of our ACRI implementation for classifying documents with recurrent items to the ARC-BC approach using the exact same setup, we provided the executable application with flag parameters indicating whether reoccurrence of words in the documents is to be considered or not. In other words we can also simulate the ARC-BC algorithm as in [12] but with the same setup as for ACRI. Although ACRI program suite contains the ARC-BC approach, from now on the term ACRI will be used only to denote the method with recurrent items.

At first, we tested both approaches using relatively high support. We produced several different sets of rules to be used in the classifier. For ARC-BC we chose the support threshold range from 10 to 30% with the step of 5%; and 15 to 65% with the same step for our approach. The difference between the support thresholds lies in the definition of support for mining rules with recurrent items. As we mentioned in section 4.1 a single transaction (document) can support an itemset (a set of words) more than once. Therefore, if we consider support as the ratio of support count to the total number of transactions, as it was introduced

in [9], we may encounter support more than 100% for some itemsets. On the other hand, if we choose the definition presented in [13], i.e., the ratio of support count to the number of distinct items (words), the support will never reach 100% as long as there is more than one distinct item in the dataset (which is quite obvious). Actually, in practice, the latter support definition requires for setting very small thresholds to obtain reasonable results. Hence, we decided to use the first one as it is more similar to the "classical" definition of support. It is important to notice that no matter which definition we choose, it eventually leads to setting the same support count with ARC-BC (i.e., the absolute number of transaction supporting an itemset).

For each support threshold we set three different confidence thresholds: 0, 35, and 70%. The latter threshold was used in [12] as minimum reasonable threshold for producing rules; the first one (no threshold) was introduced to observe the reaction of the classifier for dealing with a large number of rules; and the threshold of 35% is simply the middle value between the two others. For each single experiment we tried to keep the level of more then 98% of classified objects, which resulted in manipulating the *coverage CV* (see section 4.2) from 0.3 to 1. We discarded cases for which it was not possible to set *CV* to satisfy the minimum number of classified objects. More than 90% of the remaining results had $CV = 1$. We also performed experiments without specifying *CV* (using different methods of choosing applicable rules); however, they eventually produced lower accuracy than those with specified $CV > 0.3$. We used different classification techniques for choosing the most applicable rule matching the object. *Best confidence* and *dominant matching class* matching methods were utilized for both ARC-BC and ACRI approaches. Additionally, ACRI was tested with the *cosine measure* technique. So for all experiments herein reported the *coverage* (*CV*) is set to 1. In other words, for a rule to be selected for classification, all features expressed in the antecedent of the rule have to be observed in the new object to classify.

We also performed tests with combination of matching techniques with different tolerance factors for each test. An example scenario, reported in Figure 6, combines cosine measure, dominant matching class and best confidence. The result in Figure 6 follows the scenario: (1) choose top 20% of rules with the best cosine measure, then (2) choose 50% of the remaining rules with the highest confidence, and then (3) choose the rule based on the dominant class technique.

We also did a battery of tests using relatively low supports. This significantly increases the number of classification rules. We varied the support between 0 and 0.1% and compared the harmonic average of precision and recall (F1 measure) for the same cases as before: *Best confidence* and *dominant matching class* for both ARC-BC and ACRI approaches, and the *cosine measure* technique for ACRI.

Given the true positives (TP), the true negatives (TN), the false positives (FP), and the false negatives (FN) from the confusion matrix resulting from classifying a test set, precision, recall and F1 are defined as follows: $P = \frac{TP}{TP + FP}; R = \frac{TP}{TP + FN}; F1 = \frac{2PR}{P + R}.$

5.3 COMPARATIVE RESULTS

Categorizing documents from the Reuters dataset was best performed when the confidence level of the rules was at the 35% threshold for both the ACRI and ARC-BC approaches. For ARC-BC classifier, the best strategy was to use dominant factor, whereas in case of ACRI

combination of cosine measure and confidence factors worked best. Figure 3 shows the relationship between support and accuracy for these approaches. Comparing the best-found results, ARC-BC slightly outperforms the ACRI using the dominant matching class strategy at the 20% support level. However, ARC-BC seems to be more sensitive to changes of the support threshold. The accuracy of ACRI virtually does not depend on the support threshold and is stable as can be seen in Figure 3. In the case of ARC-BC the accuracy decreases significantly when this support is greater than 20%.

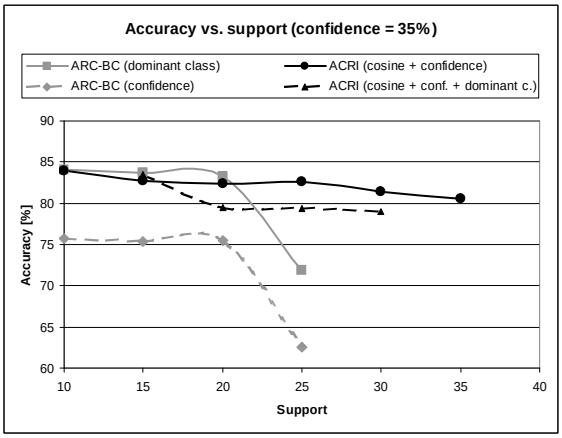

Figure 3 Accuracy of the ACRI and ARC-BC with high supports

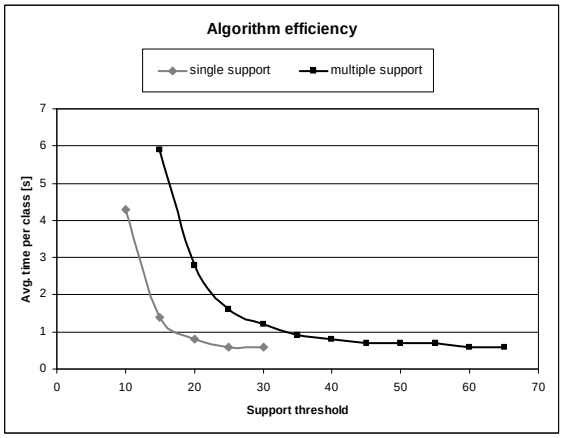

Figure 4: Algorithms CPU time efficiency

Figure 5 and Figure 6 show the number of generated rules with and without recurrent items. As it can be observed, considering recurrences results in having more rules, this has its origin in different support definition. The other interesting relationship is that by increasing the confidence threshold from 0% to 35%, the difference between number of rules decreases more rapidly for ACRI.

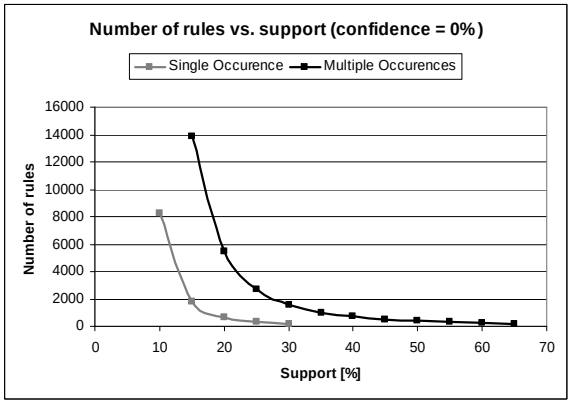

Figure 5 Number of rules with confidence = 0%

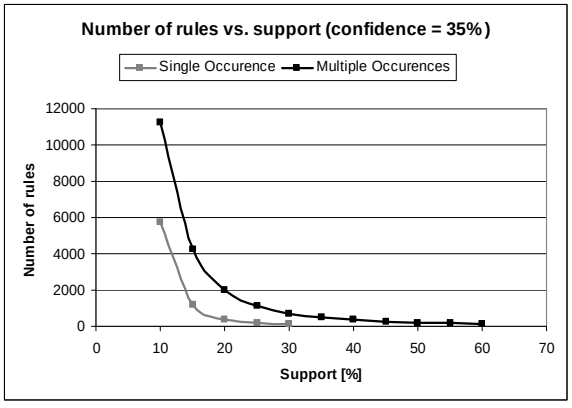

Figure 6 Number of rules with confidence = 35%

Experiments using low support thresholds confirm the stability of ACRI with regard to support. When varying the support from 0% to 0.1% ARC-BC loses in precision and recall

while ACRI remains relatively consistent or looses effectiveness on a slower pace. Figure 6 also shows that ACRI outperforms ARC-BC at these lower support thresholds. Using the cosine measure for selecting rules appears to be the best strategy. The cosine measure is also the best rule selection strategy when considering the number of rules discovered. In addition, the more rules are available the more effective the cosine measure becomes at selecting the right discriminant rules (Figure 6).

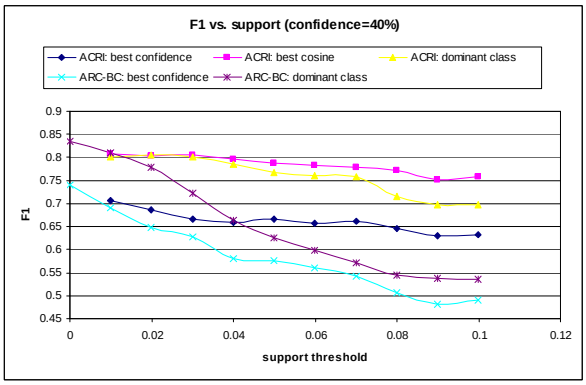

Figure 7: Effectiveness at low support.

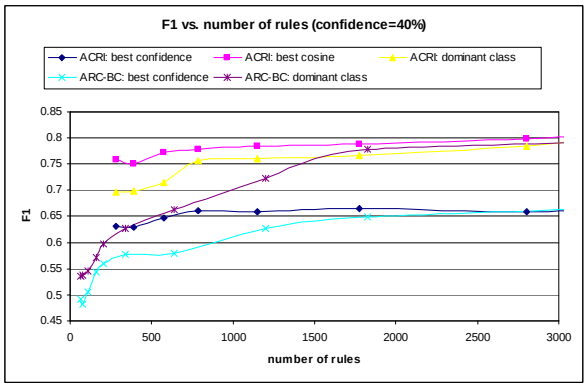

Figure 8: Effectiveness versus size of model

Figure 4 shows the relationship between running time for rule generator with and without considering recurrent items. The algorithm with recurrences is slower, since it has to search a larger space, yet the differences become smaller when increasing the support threshold.

The best results for ACR-BC were found in [12] for confidence threshold greater than 70%. However, our experiments show that effectiveness is better on lower confidence for both ARC-BC and ACRI approaches. In other words, some classification rules with low confidence have more discriminant power and are selected by our rule selection strategies. This discrepancy with previous results may be explained by the use of the different method of counting support and confidence or/and by the fact that our classifier ACRI with re-occurring items and without re-occurrence consideration to simulate ARC-BC is using a different setup for rule selections.

6 CONCLUSIONS AND FUTURE WORK

In this paper we introduced the idea of combining associative classification and mining frequent itemsets with recurrent items. We combined these two and presented ACRI, a new approach of associative classification with recurrent items. We also suggest new strategies to select classification rules during the classification phase. In particular, using the cosine measure to estimate the similarity between objects to classify and available rules is found very effective for associative classifiers that consider re-occurrence. When comparing our ACRI approach with other associative classifiers represented by ARC-BC we found that considering repetitions of observed features is beneficial. In particular in the case of text categorization, repetition of words has discriminant power and taking these repetitions in consideration can generated good classification rules. Our experiments also show that ACRI becomes more effective as the number of rules increases in particular with our cosine measure for rule

selection. Moreover, ACRI seems to be less sensitive, with respect to accuracy, to the support threshold, while other associative classifiers are typically very sensitive to the support threshold which is very difficult to determine effectively in practice. This research is still preliminary. We intend to investigate the possibility to eliminate the need for the support threshold by automatically selecting an optimal support based on available data. This is in part possible because ACRI is not substantially sensitive to the variation of the support. We are also investigating other rule selection strategies since selecting the right rules has a paramount effect on the precision of a classifier. Moreover, pruning the large set of classification rules can improve the accuracy and speed of the classifier.

REFERENCES:

[1] Agarwal R., Srikant R., Fast Algorithms for Mining Association Rules, *International Conference Very Large Data Bases,* pp. 487-499, 1994

[2] Antonie M.-L., Zaïane O., Coman A., Associative Classifiers for Medical Images, Lecture Notes in Artificial Intelligence 2797, *Mining Multimedia and Complex Data,* Springer-Verlag, pp. 68-83, 2003

[3] Antonie M.-L., Zaïane O., Coman A., Mammography Classification by an Association Rule-Based Classifier, *Third International ACM SIGKDD Workshop on Multimedia Data Mining (MDM/KDD'2002)*, pp. 62-69, 2002

[4] Han J., Kamber M., Data Mining: Concepts and Techniques, *Morgan Kaufmann Publishers*, 2001

[5] Han J., Pei J., Yin Y., Mining Frequent Patterns without Candidate Generation, ACM SIGMOD Intl. Conference on Management of Data, 2000

[6] Li W., Han J., Pei J., CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules, *IEEE International Conference on Data Mining*, pp. 369-376, 2001

[7] Liu B., Hsu W., Ma Y., Integrating Classification and Association Rule Mining, *Knowledge Discovery and Data Mining*, pp. 80-86, 1998

[8] Liu B., Hsu W., Ma Y., Integrating Classification and Association Rule Mining, *Knowledge Discovery and Data Mining*, pp. 80-86, 1998

[9] Ong K.-L., Ng W.-K., Lim E.-P., Mining Multi-Level Rules with Recurrent Items Using FP'-Tree, *ICICS*, 2001

[10] Wang W., Yang J., Yu P., WAR: Weighted Association Rules for Item Intensities, *Knowledge and Information Systems*, vol. 6, pp. 203-229, 2004

[11] Zaïane O., Han J., Finding Spatial Associations in Images, *Data Mining and Knowledge Discovery: Theory, Tools, and Technology II, SPIE 14th International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, pp. 138-147, 2000

[12] Zaïane O., R., Antonie M., L., Classifying Text Documents by Associating Terms with Text Categories, *Thirteenth Australasian Database Conference (ADC'02)*, pp. 215-222, 2002

[13] Zaïane O., R., Han J., Zhu H., Mining Recurrent Items in Multimedia with Progressive Resolution Refinement, *International Conference on Data Engineering (ICDE'2000)*, pp. 461-470, 2000

[14] Porter's Stemming Algorithm. http://www.tartarus.org/~martin/PorterStemmer/

[15] Reuters-21578 Top 10 Topics Collection. http://www.jihe.net/datasets.htm - rttop10

[16] UCI Machine Learning Repository. http://www.ics.uci.edu/~mlearn/MLRepository.html.